EXHIBIT L

RELIABILITY ANALYSIS OF A COMPUTER SYSTEM FOR A DATA COLLECTION APPLICATION

Andrew Reibman
AT&T Bell Laboratories
Holmdel, NJ 07733

Abstract

Reliability is an important objective in many computer system applications. This paper presents a reliability modeling case study, the analysis of computer system for a data collection application. We develop a model for the system, predict its reliability, and evaluate the effectiveness of disk mirroring as a potential reliability improvement.

1 Introduction

Reliability is an important objective for many computer system applications. In this paper, we describe the reliability analysis of computer system for a data collection application. A high-level picture of the system is shown in Figure 1. The system is built around a central mini-computer platform that serves as a database manager and transaction processor. The components of the platform include the processor, memory, I/O cards, power supply, and the disks and disk controller. Users have workstations that are connected to the central processor though a local area network (LAN). The central processor also has connections to different test and data collection points located in a wide area network (WAN).

The goals of our analysis were to:

- Predict whether the system hardware will meet its reliability objective.
- Identify the main causes of outage in the system, the *reliability bottlenecks.*
- Evaluate the effectiveness of potential improvements to the architecture.

The paper is organized as follows: In section 2, we describe and analyze some reliability models for the system. In section 3, we evaluate disk mirroring as a possible reliability improvement for the system. Although we do not discuss all the details of the analysis here, the example should provide a flavor for how reliability analysis can be used as part of system design.

2 Model

In this section, we describe the system model and its analysis. Our basic approach was to use the following process [RV90]:

1. Work with the designers to understand the architecture.
2. Based on the system's reliability objectives, define a set of metrics for analysis.
3. Gather information on subsystem reliability.
4. Construct and solve a system model.
5. Refine the model until it is believable, and it meets the needs of the designers.
6. Use the results to identify the reliability bottlenecks.
7. Evaluate potential reliability improvements.

For simplicity, in the initial pass at modeling and analysis described in this paper, we analyze only steady-state expected availability (SSA). Even when only total outages are considered, SSA is an inadequate measure. For example, SSA does not distinguish between frequent short outages and infrequent long outages. It also does not give any feeling for the variability that can be expected from system to system. When refining our analysis, our first priority is to give the designers a feeling for the variability that should be expected over time in large population of systems.

An important step in predicting availability, is to determine what constitutes an "operational" system. By looking at the architecture with designer, we were able to classify potential outages into three rough classes: complete system outage (Category A), partial system outage with some loss of system functionality (category B), and partial system outage with no loss of function but loss of service to some users (Category C). Even if we use a very simple metric, like mean time to first failure or steady-state availability, the results will depend on which of these types of event is counted as a system failure. One approach to dealing with different kinds of failures would be to combine the three classes of outage into a single metric using a performability model that captures different levels of system performance [Rei90]. But, in this application partial outages are far more tolerable than total failures, so combining different types of outage is overly conservatively. In this paper, *we will focus primarily on total failures.*

In the rest of this section, we first discuss the model of the the central processor, which is the most critical subsystem. We then discuss the communication subsystems, which although less critical than the central processor are still important. Finally, we examine other sources of total outage, including power failures and software.

CH2909-0/90/0000/0066$01.00 © 1990 IEEE

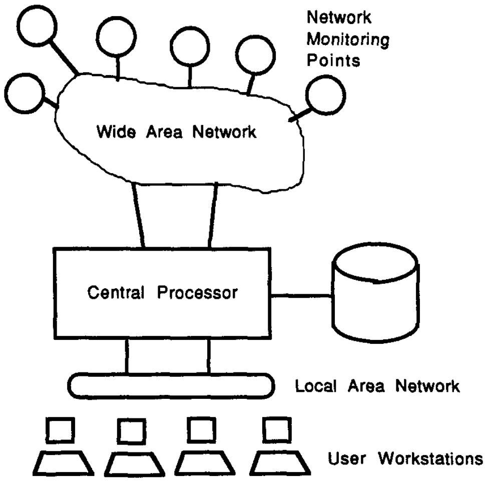

Figure 1: System Architecture

2.1 Central Processor Model

A failure of the central processor results in a complete loss of system function. It is the only subsystem that, without restriction, falls into category A. To predict the availability of the central processor, we first consider failures, then we examine repairs.

The components of the central processor are listed in Table 1, together with their failure rates. The failure of any of these components will cause the central processor to fail. The predicted total central processor failure rate is 2.2×10^{-4} failures/hour. More than half the failures are expected to come from the disk subsystem. The failure rate estimates were obtained using parts count models [KNM90]. Field experience indicates that, for systems like the central processor, these failure rate estimates are conservative, but probably within a factor of two of the actual long-term field failure rate. Even if the long-term rates are accurate, one should note that real hardware does *not* fail at a constant rate. During early life, hardware failure times often tend to follow a Weibull distribution [KNM90]; the failure rate $\lambda(t) = \lambda_s t^{-\alpha}$, with shape parameter α between .6 and .8. Thus, more than twice as many failures can be expected during the first year of operation, as would be predicted from the long-term failure rate. This *infant mortality* is an important issue that needs to be brought to the attention of both system designers and project managers.

Typically, people assume that the length of system outages corresponds to some fixed repair time, or that repairs occur at a constant rate (exponentially distributed repair times). Both of these assumptions can be deceiving, for they imply that repair times have little or no variability. The distribution of system outage times in many systems is lognormal [Ams89]. What is important to note about the lognormal distribution, is that it has a heavy tail. The variability is large; the number of long outages is greater than would be seen in an exponential distribution with the same mean. This variability needs to be considered when writing system requirements or analyzing system availability. Although we do not show the results here, we have generalized the our availability model into a semi-Markov model that includes lognormal outage times.

Subsystem	Failure Rate (failures/hour)
CPU	2.2×10^{-5}
Memory (4)	1.0×10^{-6}
Disk Drive (3)	3.3×10^{-5}
Disk Controller	2.5×10^{-5}
LAN I/O	5.5×10^{-6}
Network I/O	4×10^{-6}
Power Supply	2.5×10^{-5}
Total	2.2×10^{-4}

Table 1: Central Processor Component Failure Rates

2.2 Communication subsystems

Next, we consider the communication subsystems: the LAN connection to the users and the WAN connections to the network monitoring points.

The local area network connects users' workstations with the central processor. The predicted LAN failure rate is 10^{-5} failures/hour, and the mean time to repair is 3 hours. If the LAN fails, most users will lose service. (A few users still have direct connections to the central processor.) This is less severe than a complete failure of the central processor, but we still count it as a total system failure. In contrast, in the preliminary analysis, we ignore the failure of individual workstations.

In the complete analysis, we also examined several different LAN topologies. Some of these LAN topologies have partial failure modes that do not disconnect all of the users. For example, suppose a half of the users were served using a second LAN, that was linked to the first LAN via a bridge. Then, if the first LAN failed, all users would be disconnected. If the second LAN failed, only half the users would loose service.

The wide-area communication network connects the data collection points with the central processor. Unlike more clearly defined hardware failure rates, information on network reliability is often supplied as a value for a single end-to-end connection. The failure rate for a single connection is 5.7×10^{-4} failures/hour. The results for the entire system with several connections depend on the structure of the network and on how individual network failures affect system performance.

As a baseline, we considered a system using three data collection points with separate network connections. A conservative model would assume that all three points were required for system operation, and that any failure would result in a total system failure. This would result in more outage caused by network connections than by the central processor. Because the loss of a single data collection point is not a total system failure, this is overly conservative.

Subsystem	Failure Rate (failures/hour)	MTTR	Expected Outage (hours/year)
Central Processor	2.2×10^{-4}	4 hours	8
WAN Connection	5.7×10^{-4}	4 hours	20
LAN*	10^{-5}	3 hours	.3
Power**	7×10^{-5}	1 hour?	.6

Table 2: Subsystem Failure and Repair Rates
*Actually 10^{-5} failures/hour $+1.5 \times 10^{-6}$ failures/hour-connection.
*Assumes system has UPS with 15 minute battery backup .

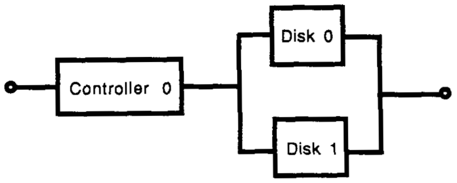

Figure 2: Reliability Block Diagram for a Mirrored Disk Subsystem with Single Disk Controller

A more realistic model assumes that a single network connection failure causes a degradation in system performance, but not a system failure. If the connections are independent (an optimistic assumption), this would be a 2-out-of-3 system, with an expected unavailability of approximately 3×10^{-5}, significantly less than the central processor. Dependence between the network connections would increase this unavailability, but it is still unlikely that these connections will be the system reliability bottleneck. More realistic models can be built to investigate how partial network outage and varying degrees of dependence between connections affect the overall system's reliability.

2.3 Other Outage Sources

In addition to the communications and central processor hardware, there are two other significant sources of outage that need to be considered, even though they are not the focus of our study: power and software.

A major source of outage that is often overlooked in reliability studies is power failures. In a study of minicomputers powered by unconditioned AC power, the median system suffered 31 failures per year from poor power conditioning (e.g. voltage dips) or power failures [GS82]. This corresponds to a failure rate of 3.5×10^{-3} failures/hour. However, 87% of these outages were due to short voltage dips, which can be filtered out using simple line voltage regulators. Power failures made up 4.7% of the power disturbances. Half of these power failure were less than 38 seconds. Even with line voltage regulators, far more outage can be expected from power failures than from hardware failures. Luckily, power quality can be improved economically using an uninterruptible power supply. Using a 15 minute uninterruptible power supply should eliminate approximately 98 % of all outages caused by poor power condition, reducing the failure rate to 7×10^{-5} failures/hour. In installations with both UPSs and generator backups (which cover power failures that are longer than 15 minutes) system outage due to power failures is almost negligible.

A second major source of outage that is of great concern is software [MIO87]. Before the system is built, it is difficult to get a reasonable estimate of software reliability. By looking at similar systems, we can get a rough idea of software reliability. Then the hardware reliability number can be used to determine a reasonable reliability budget for the software, i.e. how reliable the software must be in order not to be the system reliability bottleneck.

3 Enhancement: Disk Mirroring

The information in the previous section suggestion suggests that the disks are the major source of total outage. A potential reliability improvement is to mirror the disk drives. At first examination, one would assume that a mirrored disk system could be represented by the reliability block diagram in Figure 2. Similarly, mirrored disks with separate controllers could be represented by the block diagram in Figure 3. In fact, the situation is more complicated.

The coverage provided by mirroring the disks may be imperfect; not all disk failures may be masked by the mirroring. A detailed discussion of modeling coverage can be found in [DT89]. Some failures (like failures of the disk media) are completely masked by the mirroring, and can be repaired with the system on-line. Other failures may directly cause a system failure or may require some system outage to repair, even if the system is mirrored. To capture imperfect coverage, we can use a Markov reliability model (Figure 4). State 2 represents two disks working, State 1 represents one disk working, and State 0 represents a system failure. The arc connecting State 2 and State 0 represents failure due to imperfect coverage — a fault occurred that caused both disks to fail simultaneously.

The parameter c, the coverage, is the probability that a single fault is successfully masked by the mirrored disk. If c is much less than one, imperfect coverage will be the dominant cause of system failure. Typical values of c depend on the application. For illustration, consider a mirrored disk system with 95% coverage and only a single repair facility. The original disk subsystem without mirroring had an availability of 5×10^{-4}. Assuming perfect coverage and a single repair facility, the expected availability of the disk subsystem would be 5×10^{-7}. With 95% coverage, the expected availability of the disk subsystem would drop to 5×10^{-5}. This suggests that disk mirroring should still provide a significant reliability improvement. More detailed investigation of the actual coverage provided by mirroring can be done using more complicated models, or by examining field data from existing systems with mirrored disks.

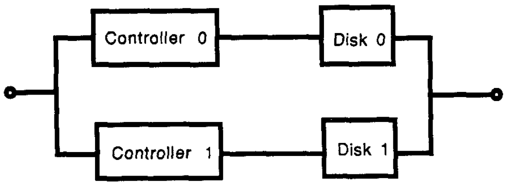

Figure 3: Reliability Block Diagram for a Mirrored Disk Subsystem with Redundant Disk Controllers

4 Conclusions

We have presented a preliminary analysis of the reliability of an architecture for a computer system for a data collection application. The analysis identified the disk subsystem as a potential reliability bottlenecks in the system. We analyzed disk mirroring as a potential reliability improvement. Refinements to our analysis include (1) using more realistic subsystem repair and outage time models, (2) obtaining more accurate subsystem failure rates, (3) a performability model for network connections, and (4) analyzing the other fault-tolerant design features redundant communication links or a distributed processor.

References

[Ams89] S. Amster. Meaningful system reliability requirements. In *Quality Improvement Techniques for Manufacturing, Products, and Services*, December 1989.

[DT89] J. Bechta Dugan and K. S. Trivedi. Coverage modeling for dependability analysis of fault-tolerant systems. *IEEE Transactions on Computers*, 38(6):775–787, 1989.

[GS82] M. Goldstein and P. Speranza. The quality of U.S. commercial AC power. In *Proceedings of INTELEC '82*, October 1982.

[KNM90] D. J. Klinger, Y. Nakada, and M. A. Menendez, editors. *AT&T Reliability Manual*. Van Nostrand Reinhold, New York, 1990.

[Law82] J. Lawless. *Statistical Models and methods for Lifetime Data*. John Wiley & Sons, New York, 1982.

[MIO87] J. Musa, A. Iannino, and K. Okumoto. *Software Reliability: Measurement, Prediction, Application*. McGraw-Hill, 1987.

[Rei90] A. Reibman. Modeling the effect of reliability on performance. *IEEE Transactions on Reliability*, 1990. To appear.

[RV90] A. Reibman and M. Veeraraghavan. Modeling computer system reliability: an overview for system designers. 1990. Submitted for publication.

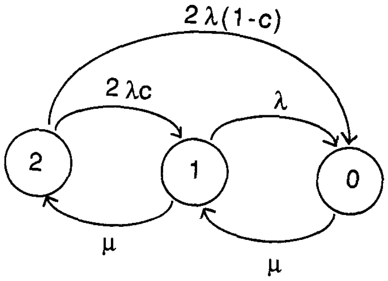

Figure 4: Markov Reliability for Disk Subsystem with Coverage